# Section 7: Simple SQL Exercises

## How many records are in the nyc\_streets table?

## How many streets in New York have names that start with ‘B’?

## What is the population of New York city?

The “nyc\_census\_blocks” table includes a “popn\_total” field which is the number of people in each block.

## What is the population of the Bronx?

The “nyc\_census\_blocks” table includes a “boroname” field which is the borough each block is in.   
‘The Bronx’ is a borough.

## How many "neighborhoods" are in each borough?

The “nyc\_neighborhoods” table includes a “boroname” field and a "name" field. Group by!

## For each borough, what percentage of the population is white?

The “nyc\_census\_blocks” table includes a “popn\_white” field that is the number of self-identified white people in each block.

## Data Dictionary

Useful columns in the “nyc\_census\_blocks” table:

|  |  |
| --- | --- |
| blkid | A 15-digit code that uniquely identifies every census block. Eg: 360050001009000 |
| popn\_total | Total number of people in the census block |
| popn\_white | Number of people self-identifying as “white” in the block |
| popn\_black | Number of people self-identifying as “black” in the block |
| popn\_nativ | Number of people self-identifying as “native american” in the block |
| popn\_asian | Number of people self-identifying as “asian” in the block |
| popn\_other | Number of people self-identifying with other categories in the block |
| boroname | Name of the New York borough: Manhattan, The Bronx, Brooklyn, Staten Island, Queens |
| geom | MultiPolygon boundary of the block |

## Useful Functions

|  |  |
| --- | --- |
| Count(\*) | An aggregate function that returns the number of records in the query. |
| GROUP BY [field] | A SQL statement the performs aggregations based on the distinct values of the provided field. |
| [field] LIKE ‘V%’ | A SQL pattern matching operator, where the field is matched against a pattern and ‘%’ is the ‘global match’ character. The example returns all field values starting with ‘V’. |
| Sum([field]) | An aggregate function that returns the total value of the field over all records in the query. |
| Avg([field]) | An aggregate function that returns the average value of the field over all records in the query. |

# Section 9: Geometry Exercises

## What is the area of the ‘West Village’ neighborhood? What are the units of your answer?

The “nyc\_neighborhoods” table includes a “name” field.

## What is the geometry type of ‘Pelham St’? How long is it?

The “nyc\_streets” table includes a “name” field.

## What is the GML representation of the ‘Broad St’ subway station? What about the KML? Why are they different?

The “nyc\_subway\_stations” table includes a “name” field.

## How many census blocks in New York City have a hole in them?

How do you represent polygons with a hole?   
You will have to use ST\_GeometryN to turn MultiPolygons into Polygons.

## What is the most westerly subway station?

Try using “ORDER” and “LIMIT”.

## What is the area of Manhattan in acres?

The “nyc\_census\_blocks” table is an area geometry and has a “boroname” column.  
There are 4047 square meters in an acre.

## Data Dictionary

Useful columns in the “nyc\_streets” table:

|  |  |
| --- | --- |
| name | Common name of the street |
| oneway | Is this a one-way street? |
| type | What kind of street is this? |
| geom | MultiLinestring geometry of the street |

Useful columns in the “nyc\_subway\_stations” table:

|  |  |
| --- | --- |
| name | Common name of the station |
| routes | Comma-separated list of routes that serve this station |
| geom | Point geometry of the street |

## Useful Functions

|  |  |
| --- | --- |
| Sum([field]) | An aggregate function that returns the total value of the field over all records in the query. |
| LIMIT n | Restrict the query to return only the first “n” rows. |
| ORDER BY [field] | Return the query in order sorted by the field. |
| ORDER BY [field] DESC ORDER BY [field] ASC | Return the query in descending/ascending order sorted by the field. |
| ST\_X(point) | Returns the X coordinate of the point |
| ST\_Y(point) | Returns the Y coordinate of the point |
| ST\_Length(geometry) | Returns the length of the geometry |
| ST\_Area(geometry) | Returns the area of the geometry |
| ST\_StartPoint(line) | Returns the first point in the line |
| ST\_EndPoint(line) | Returns the last point in the line |
| ST\_NumPoints(line) | Returns the number of vertices in a linestring |
| ST\_NumInteriorRings(polygon) | Returns the number of interior rings (holes) in a polygon |
| ST\_NumGeometries(collection) | Returns the number of sub-geometries in any geometry collection (MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION) |
| ST\_GeometryN(geometry, n) | Returns the n’th geometry in the collection (starting from 1) |
| ST\_AsGML(geometry) | Returns the GML representation |
| ST\_AsKML(geometry) | Returns the KML representation |
| ST\_AsGeoJSON(geometry) | Returns the GeoJSON representation |
| ST\_AsText(geometry) | Returns the well-known-text representation |

# Section 11: Spatial Relationship Exercises

## What is the well-known text for the street ‘Atlantic Commons’?

## In what neighborhood is POINT(586782 4504202) (aka, the start of Atlantic Commons)?

Don't forget to include SRID 26918 when you construct your geometry.

The “nyc\_neighborhoods” table includes a "name” field.

## How many people live within 50 meters of POINT(586782 4504202)?

The “nyc\_census\_blocks” table includes a “popn\_total” field.

## For ‘LINESTRING(0 0, 2 2)’ and ‘POINT(1 1)’ which of these relationships are true? Intersects, Touches, Contains, Disjoint, Overlaps, Crosses, Within.

A sub-query might make this terser to type.

## How far apart are ‘Columbus Cir’ and ‘Fulton Ave’?

For SQL experts only!

## Useful Functions

|  |  |
| --- | --- |
| Sum([field]) | An aggregate function that Returns the total value of the field over all records in the query. |
| ST\_Contains(A, B) | Returns true if geometry A contains geometry B |
| ST\_Crosses(A, B) | Returns true if geometry A crosses geometry B |
| ST\_Disjoint(A, B) | Returns true if the geometries do not “spatially intersect” |
| ST\_Distance(A, B) | Returns the minimum distance between geometry A and geometry B |
| ST\_DWithin(A, B, d) | Returns true if geometry A is distance or less from geometry B |
| ST\_Equals(A, B) | Returns true if geometry A is the same as geometry B |
| ST\_Intersects(A, B) | Returns true if geometry A intersects geometry B |
| ST\_Overlaps(A, B) | Returns true if geometry A and geometry B share space, but are not completely contained by each other |
| ST\_Touches(A, B) | Returns true if geometry A and geometry B share space, but are not completely contained by each other |
| ST\_Within(A, B) | Returns true if geometry A is within geometry B |

# Section 13: Spatial Joins Exercises

## What subway station is in the ‘Little Italy’ neighborhood?

## What are all the neighborhoods served by the 6 train?

The “nyc\_subway\_stations” table has a “routes” column with values like ‘A,C,6’ and ‘5,6,Q’.

## After 9/11, the ‘Battery Park’ neighborhood was off limits for several days. How many people had to be evacuated?

The “nyc\_census\_blocks” table has a “popn\_total” column.

## What neighborhood has the highest population density (persons/km2)? The lowest?

There are 1,000,000 m2 in a km2.

## Useful SQL

The pattern for a spatial join is commonly

SELECT a.field, b.field  
FROM table\_a AS a  
JOIN table\_b AS b  
ON ST\_Something(a.geom, b.geom)  
WHERE a.field = ‘SOMETHING’;

For spatial joins that aggregate results over the whole set, the pattern is commonly

SELECT Sum(a.field), b.field  
FROM table\_a AS a  
JOIN table\_b AS b  
ON ST\_Something(a.geom, b.geom)  
WHERE a.field = ‘SOMETHING’

GROUP BY b.field;

Note the aggregate function around one term and the grouping on the other.

## Useful Functions

|  |  |
| --- | --- |
| strpos(str, char) | Returns the character position of the character in the string or 0 if the character is not in the string. |
| field LIKE ‘%thing%’ | Returns true if ‘thing’ is appears within a string in the field |
| Sum(field) | An aggregate function that Returns the total value of the field over all records in the query. |
| ST\_Area(geometry) | Returns the area of the geometry in square units |
| ST\_Contains(A, B) | Returns true if geometry A contains geometry B |
| ST\_Crosses(A, B) | Returns true if geometry A crosses geometry B |
| ST\_Disjoint(A, B) | Returns true if the geometries do not “spatially intersect” |
| ST\_Distance(A, B) | Returns the minimum distance between geometry A and geometry B |
| ST\_DWithin(A, B, d) | Returns true if geometry A is distance or less from geometry B |
| ST\_Equals(A, B) | Returns true if geometry A is the same as geometry B |
| ST\_Intersects(A, B) | Returns true if geometry A intersects geometry B |
| ST\_Overlaps(A, B) | Returns true if geometry A and geometry B share space, but are not completely contained by each other |
| ST\_Touches(A, B) | Returns true if geometry A and geometry B share space, but are not completely contained by each other |
| ST\_Within(A, B) | Returns true if geometry A is within geometry B |

# Section 16: Projection Exercises

## What is the SRID of the nyc\_streets table? What projection does that SRID represent?

## What is the total length of all streets in New York, as measured in UTM 18N?

See the question above!

## What is the total length of all streets in New York, as measured in Stateplane Long Island (EPSG 2831)?

Why is the answer different from above?!

## How many distinct streets cross the 74th meridian?

The data run from about -73.5 to -74.5 longitude and 40.5 to 40.9 latitude.  
Streets can have multiple rows with the same name.

## Useful Metadata

The “spatial\_ref\_sys” table contains all the information about SRID values in the database. It is user-configurable, but is loaded by default with all the values from the EPSG database.

|  |  |
| --- | --- |
| srid | Unique identifier for this spatial reference system |
| auth\_name | Authority that defined this system (usually EPSG) |
| auth\_srid | Identifier used by the authority (note that this does **not** have to be the same as the value used for the internal database SRID) |
| srtext | “well known text” definition of the spatial reference system. The same format as used in “prj” files. |
| proj4text | Proj4 text definition of the spatial reference system. Used by the PostGIS re-projection engine in ST\_Transform calls. |

## Useful Functions

|  |  |
| --- | --- |
| DISTINCT | Returns only unique rows (all values the same) |
| Sum(field) | An aggregate function that Returns the total value of the field over all records in the query. |
| ST\_Transform(A, srid) | Returns a geometry in the requested spatial reference system, transforming the coordinates of the geometry. |
| ST\_SRID(A) | Returns the SRID value of the geometry |
| ST\_SetSRID(A, srid) | Updates the SRID value of a geometry, does not alter the coordinates. |
| ST\_GeomFromText(text, srid) | Create a new geometry from a well-known text form and set the SRID value at the same time. |
| ST\_XMin(A) | Returns the minimum value of the X coordinate of a geometry. |
| ST\_XMax(A) | Returns the maximum value of the X coordinate of a geometry. |
| ST\_Intersects(A, B) | Returns true of geometry A and geometry B intersect, false if they are disjoint. |

# Section 17: Geography Exercises

## How far is New York from Seattle? What are the units of the answer?

New York = POINT(-74.0064 40.7142) , Seattle = POINT(-122.3331 47.6097)

## What is the total length of all streets in New York, calculated on the spheroid?

You can cast geometry to geography using the “::geography” syntax or the “geography(geometry)” function.

## Does ‘POINT(1 2.0001)’ intersect with ‘POLYGON((0 0, 0 2, 2 2, 2 0, 0 0))’ in geography? In geometry? Why the difference?

ST\_GeomFromText and ST\_GeogFromText are helpful here.

## Casting Types

You can convert values from one PostgreSQL type to another using the “::type” syntax.

-- Converts text to geometry

SELECT ‘POINT(0 0)’::geometry;

-- Converts text to geography to geometry

SELECT ‘POINT(0 0)’::geography::geometry

## Useful Functions

|  |  |
| --- | --- |
| Sum(field) | An aggregate function that Returns the total value of the field over all records in the query. |
| ST\_Transform(geom, srid) | Returns a geometry in the requested spatial reference system, transforming the coordinates of the geometry. |
| ST\_GeomFromText(text, srid) | Create a new geometry from a well-known text form and set the SRID value at the same time. |
| ST\_Distance(geog1, geog2) | Returns the shortest distance in meters between two geographies. |
| ST\_Length(geog) | Returns the length in meters of a geography. |
| ST\_GeogFromText(text) | Create a new geography from a well-known text form. |

# Section 18: Geometry Construction Exercises

## How many census blocks don’t contain their own centroid?

## Union all the census blocks into a single output. What kind of geometry is it? How many parts does it have?

## What is the area of a one unit buffer around the origin? How different is it from what you would expect? Why?

The value of π can be accessed with the pi() function.  
There is a three-parameter variant of ST\_Buffer(geom, radius, n) What do you think the third parameter controls?

## The Brooklyn neighborhoods of ‘Park Slope’ and ‘Carroll Gardens’ are going to war! Construct a polygon delineating a 100 meter wide DMZ on the border between the neighborhoods. What is the area of the DMZ?

Hint. A self-join will be required. Buffer and then intersect.

## Useful Functions

|  |  |
| --- | --- |
| Count(field) | An aggregate function that returns the number of rows in the query result set. |
| ST\_Intersection(A, B) | Returns a geometry that covers the area shared by geometries A and B. |
| ST\_Centroid(A) | Returns a point at the center of mass of geometry A. |
| ST\_PointOnSurface(A) | Returns a point guaranteed to be within the boundary of geometry A. |
| ST\_Union(A, B) | Returns a geometry that covers all the area covered by geometry A or geometry B. |
| ST\_Contains(A, B) | Returns true if geometry A contains geometry B. |
| ST\_Union([geometryset]) | Aggregate function returns a geometry that covers all the area covered by all elements of the input geometry set. |
| ST\_GeometryType(A) | Returns the geometry type of geometry A. |
| ST\_NumGeometries(A) | Returns the number of sub-geometries in a geometry collection. |
| Pi() | Returns the value of pi. |
| ST\_Buffer(A, d) | Returns a polygonal geometry that covers all the area within a radius d of geometry A and all the area within A itself. |

# Section 20: Validity Exercises

## Create an invalid geometry. Verify it is invalid and why the system thinks it is invalid.

The rules: Interior rings inside exterior, no ring intersections except at a single point, no ring self-intersections, no multi-polygons with intersecting parts.  
Here is some graph paper to make it easier!

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Try to repair your invalid geometry. What is the result?

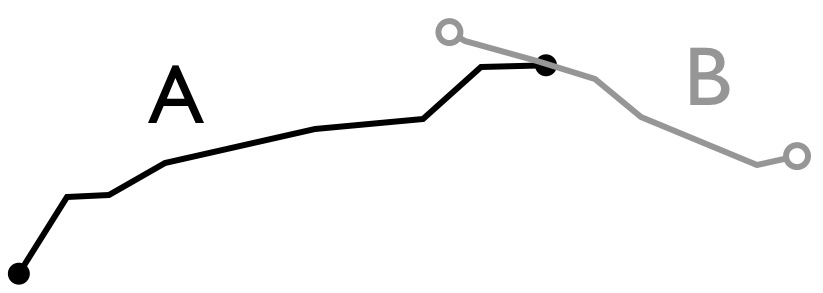
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Useful Functions

|  |  |
| --- | --- |
| ST\_GeomFromText(wkt, srid) | Create a geometry from a well-known text representation. |
| ST\_IsValid(geometry) | Returns true if the geometry is constructed according to validity rules. |
| ST\_IsValidReason(geometry) | Returns a string describing the invalidity of a geometry or NULL if the geometry is valid. |
| ST\_IsValidDetail(geometry) | Returns a composite object including a validity string and a point location where the invalidity occurs, or NULL if the geometry is valid. |
| ST\_Buffer(geometry, 0) | Rebuilds (often) a valid version of an invalid input polygon. |
| ST\_Perimeter(geometry) | Returns the total length of line work in a polygon boundary, unaffected by invalidities. |
| ST\_Area(geometry) | Returns the area of a geometry, may be affected by invalidity (in particular inverted ring sections). |

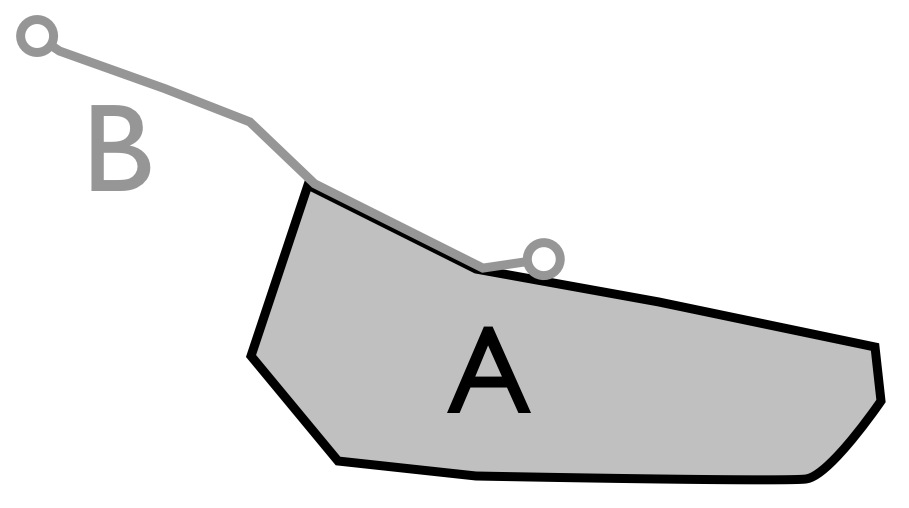
# Section 23: DE9IM Exercises

## Fill in the DE9IM matrix for this geometry interaction.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | B | | |
|  |  | I | B | E |
| A | I |  |  |  |
| B |  |  |  |
| E |  |  |  |

## Fill in the DE9IM matrix for this geometry interaction.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | B | | |
|  |  | I | B | E |
| A | I |  |  |  |
| B |  |  |  |
| E |  |  |  |